A Novel Production Scheduling Approach Based on Improved Hybrid Genetic Algorithm

Lili Dai[1], He Lu[1,2], Dezheng Hua[2], Xinhua Liu[2], Hongming Chen[2], Adam Glowacz[3], Grzegorz Królczyk[4], Z Li[4,*]

[1] Lianyungang Normal College, Lianyungang 222006, China
[2] School of Mechatronic Engineering, China University of Mining and Technology, Xuzhou 211006, China
[3] Department of Automatic, Control and Robotics, AGH University of Science and Technology, 30-059 Kraków, Poland
[4] Faculty of Mechanical Engineering, Opole University of Technology, Opole 45-758, Poland
* Correspondence: z.li@po.edu.pl

Abstract: Due to the complexity of the production shop in discrete manufacturing industry, traditional genetic algorithm (GA) cannot solve the production scheduling problem well. In order to enhance the GA-based method to solve the production scheduling problem, the simulated annealing algorithm (SAA) is used to develop an improved hybrid genetic algorithm. Firstly, the crossover probability and mutation probability of the genetic operation are adjusted, and the elite replacement operation is adopted for simulated annealing operator. Then, a mutation method is used for the comparison and replacement of the genetic operations to obtain the optimal value of the current state. Lastly, the proposed hybrid genetic algorithm is compared with several scheduling algorithms, and the superiority and efficiency of the proposed method are verified in solving the production scheduling.

Keywords: production scheduling, hybrid Genetic algorithm, artificial intelligence, sustainable design, discrete manufacturing

1. Introduction

Production scheduling is one of the classic non-deterministic problems, involving aircraft carrier scheduling, port cargo scheduling, parts processing scheduling and many other fields [1-2]. In the production field, the scheduling of production workshops, equipment, personnel and materials is different for different manufacturing enterprises, and the production cycle is also varied [3-5]. With the emergence of large-scale systems for production and the proposal of a variety of intelligent algorithms, the production scheduling of workshop has attracted more extensive attention from managers, and remarkable results have been achieved continuously [6]. In the competitive environment of new era, enterprises have played more attention to how to use production scheduling to quickly realize resource allocation, deal with production schedule and improve production efficiency [7].

Although there have been many intelligent scheduling algorithms, it is still an important goal to seek more efficient and practical scheduling approaches in the production scheduling of discrete manufacturing industry [8-9]. Considering the complexity and diversity of discrete production scheduling, there are two algorithms to solve the problems [10-13]. One is based on the traditional unified research, which mainly includes Lagrange relaxation method [14], branch-and-bound method [15] and mathematical programming method [16], etc., that is not effective in practical engineering at present. The other is heuristic algorithm, mainly including genetic algorithm, simulated annealing algorithm [17], particle swarm optimization algorithm [18] and ant colony algorithm [19], etc. For example, Choi et al. [20] proposed a mixed integer programming model and local search algorithm to solve project scheduling in a variety of manufacturing environments. Nie et al. [21] studied the dynamic scheduling problems of job publication by date, proposed a heuristic algorithm and a reactive scheduling strategy based on gene expression coding, and applied it to the production scheduling. These algorithms are

simple in structure, easy to implement, which can achieve satisfactory results in solving production scheduling [22].

Among many intelligent algorithms, genetic algorithm (GA) with special optimization mode is an effective approach to solve production scheduling problems, and has achieved some results [23-24]. Pezzellaa et al. [25] proposed a GA that combines multiple strategies for initial population generation, selection and reproduction, which is used to solve production scheduling problems. Giovanni and Pezzella [26] proposed an improved GA for distributed production scheduling problems, which determines the processing path of the workpiece through a greedy decoding process. Zhang et al. [27] used pareto-optimization-based GA and two new objective functions based on setup and synergy costs to solve production scheduling problems. Chamnanlor et al. [28] proposed a hybrid GA based on ant colony algorithm for production scheduling. Meng Yue et al. [29] created a hybrid algorithm of path reconnection for the production scheduling problems, which combined genetic algorithm, domain structure algorithm and path reconnection algorithm to further enhance the calculation ability. Zhao et al. [30] proposed a hybrid genetic simulated annealing algorithm in the production scheduling, where the problem of algorithm prematurity is solved, but the rapid and substantial drop in temperature would easily make the search data incomplete, leading to the loss of good individuals. Wang et al. [31] used the hybrid genetic simulated annealing algorithm to deal with the production scheduling of flexible jobs. Although simulated annealing factors were added to improve the performance of the algorithm and make the algorithm jump out of local optimal, the influence of adaptive crossover and mutation probability on the final convergence of the algorithm was not considered.

According to above literature analysis, to better and more effectively solve the production scheduling problems, an improved hybrid genetic algorithm (IHGA) is proposed in this work. Different from other hybrid genetic algorithm in selecting individuals to crossover and mutation, the proposed algorithm adopts an adaptive strategy to adjust size of probability. In the early stages of iterative at first, higher probability is used to choose more individual, expand the scope of the late optimization, increase species diversity, and reduce the probability of premature phenomena. Premature algorithm is beginning to choose a large amount of high fitness individuals, resulting in getting local optimal value quickly and not the global optimal value. In the later stage of population iterative reproduction, excellent individuals can be preserved with a low probability, and then the convergence of the algorithm relative to the optimal value can be completed as soon as possible. In simulated annealing factor, using the memory function, the optimal solution is conducted to mutate by probability to get as much as possible better individuals. A heating strategy is added to avoid falling into local optimum, and finding out the new individual compared with the fitness function of original individual.

The rest of this paper is organized as follows. Section 2 presents system modeling of production scheduling. Section 3 describes the operation process of improved hybrid genetic algorithm. In Section 4, experimental analysis is carried out. The application testing of the improved approach is carried out in Section 5. The conclusions and future works are summarized in Section 6.

2. Modelling Production Scheduling

The problems of production scheduling in discrete manufacturing industry usually refer to the determination of the processing sequence of each workpiece under various production requirements and processing constraints, and considering the relevant parts in the assembly plan. In view of the characteristics of discrete manufacturing industry and actual demand, time arrangement is the main factor. The production scheduling of discrete manufacturing industry can be described as: A batch of parts are needed to produced; The number of the parts is n; Each part contains many working procedures;

The batch of the parts are completed by using p sets of machines in the shortest possible time. The main constraint conditions for production scheduling are as follows:

(1) At the beginning of production, each part can be randomly selected and processed on the designated machine.

(2) Each equipment used for production in the workshop can only process one part at any time.

(3) Each workpiece can only be processed once or not on each equipment.

(4) Sudden interruption is forbidden after it has started.

(5) Any part in the first process is not in order, but the same part in production has a certain sequence constraint, and absolutely cannot be changed.

(6) The production of part must conform to the actual process line, and need to be practical.

(7) The processing time of each part process has been determined, and does not change with the sorting.

(8) Auxiliary time such as tool installation and part transportation is not considered.

According to the actual production situation of discrete manufacturing industry, each part has a certain entry point, but only after the production of all the relevant parts can be assembled. In order to improve the production and assembly efficiency of parts, it is necessary to reduce the overall production time of the batch of parts to a relatively minimum. Mathematical modeling is to find the completion time of the whole batch of parts, that is, the latest time to find the completion of the last parts. The mathematical model is as follows:

In terms of the production characteristics of part in workshop of discrete industry, it can be concluded that the calculation formula for the latest processing time of process P of part ji is as follows:

$$S_{j_iP} = \begin{cases} mWT_M \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (P=0) \\ \max(E_{j_iP-1}, mWT_M) \ldots\ldots\ldots\ldots\ldots\ldots (P>0) \end{cases} \quad (1)$$

The processing time of machine M is:

$$mWT_M = S_{j_iP} + T \quad (2)$$

The end time of the part j_i for process P is:

$$E_{j_iP} = S_{j_iP} + T \quad (3)$$

The latest production time for finishing the last step of part is:

$$\max_{0 \le i < m}(mWT_i) \quad (4)$$

The meaning of each variable in the mathematical model is shown in Table 1. The mathematical model describes the production scheduling, through the model the production time of each plan can be calculated, and then the time consumption of each plan can be compared, so as to select the best scheduling plan with the shortest time.

3. Improved Hybrid Genetic Algorithm for Production Scheduling

3.1. Chromosomal Coding

There are many encoding methods of chromosomes in genetic algorithm, and an appropriate encoding method can improve the efficiency and ability of finding the global optimal solution. Due to the complexity and particularity of discrete production scheduling, A procedure-based coding method is chosen in this work, and the encoding mode of chromosomes is shown in Figure 1. The length of chromosomes is related to the number of machines and parts. So, if there are m machines, n parts and process i ($0 \le i < n$) processing procedures of each machine, the length of the chromosome is chSize ≤ (n * m). Since the number of processes in the processed parts may not be equal, and the number of processes in some parts is less than the number of machines, the length of chromosome

should be the sum of the number of processes in all parts to be processed. In specific chromosome, coding rule is that the number of times a part appears on a chromosome indicates the number of processes it needs to be processed, such as the first occurrence of j_1 in chromosome means the first process of j_1 part, and the fourth occurrence means the fourth process of j_1, based on which the process of the other workpiece can be deduced.

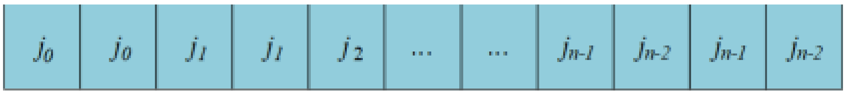

Figure 1. Discrete scheduling chromosome encoding scheme

Table 1. Variables in the mathematical model.

Name	Meaning of variable
S_{ij}	Latest start time of process j for part i
E_{ij}	Completion time of process j for part i
M_{ij}	Number of the machine required for the part i to perform process j
T_{ij}	Time taken to complete process j for part i
P_{ij}	Process of processing part i using machine j
mWT_i	Machine processing time with serial number i
P_i	Process of machining part with serial number i
j_i	The part i
P	Current operation P_{ji} of j_i
M	Number of machines P
T	Processing time of P by machine M

3.2. Design of Fitness Function

In the process of genetic algorithm optimization, fitness function is mainly used to evaluate the advantages and disadvantages of chromosomes, and it is the only condition to identify the individual quality of population. Under the condition of minimizing the completion time in production scheduling in discrete manufacturing industry, the value of fitness function is the maximum completion time obtained in accordance with the mathematical model created above, and the fitness function can be used to effectively determine the level of chromosome. In this method, the larger the fitness function value is, the longer the completion time of all the part is, and the smaller the fitness of chromosome is, the easier it is to be eliminated in the selection operation. The smaller the fitness function value is, the shorter the completion time of all the workpiece is, and the larger the fitness of chromosome is, the easier it is to be selected and inherited to the next generation population.

3.3. Selection Operation

When selecting individuals in a population, excellent individuals are selected with a high probability, while those with low fitness are selected with a low probability. So, the population will evolve in a good direction at the beginning of iteration. Through comparative analysis, the tournament method is used when selecting individuals in this work. Main operation method is to randomly select Z individuals from the population, let them compete for fitness, and then select the best one from them. In this paper, all individuals participating in the championship are the whole population, and Y is set to 3,

that is, 3 random individuals are selected from the population, and then the optimal individual j is selected. The steps are shown in Figure 2.

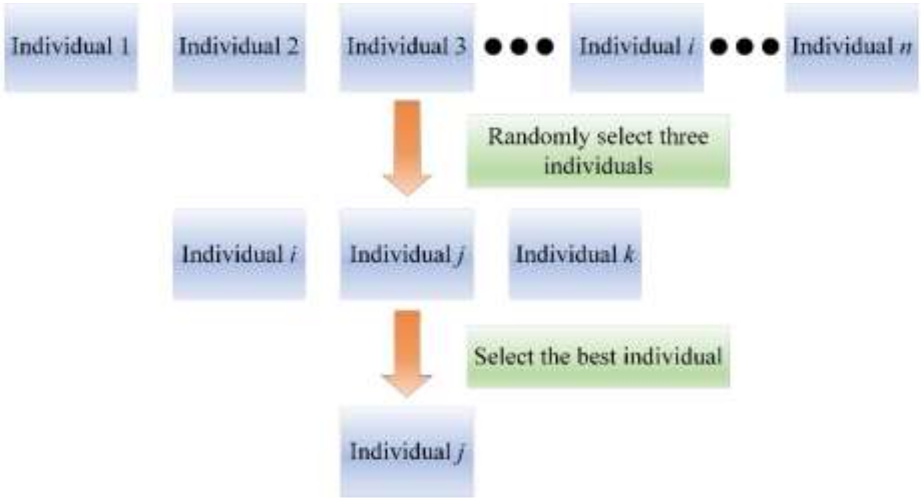

Figure 2. Tournament selection of excellent individuals.

There are usually three main ways to select individuals in genetic algorithms. The first is more commonly used tournament method, the operation method as described above. The second is roulette selection method, mainly through the comparison of individual fitness value in the population to determine the probability of individual selection, according to probability to determine the composition of offspring population. For solving the time minimization problem of production scheduling, it is need to transform fitness function into maximization problem, where the fitness value of each individual is obtained separately, and the fitness value of each individual is divided by the sum of the fitness values of all individuals. The result is the probability of individual selection. The cumulative probability of all individuals is constituted into a roulette wheel, and a new generation population is continuously obtained by generating random numbers between [0, 1] for roulette selection. The third random traversal sampling method has the same probability of individual selection as the second individual selection method. The difference is that to meet the requirement of equidistant individual selection, for example, m individuals need to be selected, the distance of the selection pointer should be 1/m, and the position of the first pointer can be determined by generating a random number in [0, 1/m]. Through the analysis of the above methods, the advantages of tournament selection individuals are small time complexity, easy parallel processing and less premature phenomenon.

3.4. Crossover Operation

One of the key operations in the continuous evolution of population is crossover operation. Crossover operation changes gene sequence of part on the basis of preserving chromosome gene fragments, which increases the diversity of population, improves the search ability of algorithm, and increases the probability of excellent individuals. Different from the previous crossover operation, an adaptive adjustment strategy is adopted in this work to keep the diversity of individuals at the initial stage of population iteration, and ensure the convergence of optimization results as soon as possible at the later stage. An OX crossover method based on process coding is adopted in the hybrid algorithm. The operator crossover diagram is shown in Figure 3 and Figure 4. In the crossover process, the starting and ending positions of chromosomes of the parent and the mother are random, so the randomness of newly generated individuals is greater.

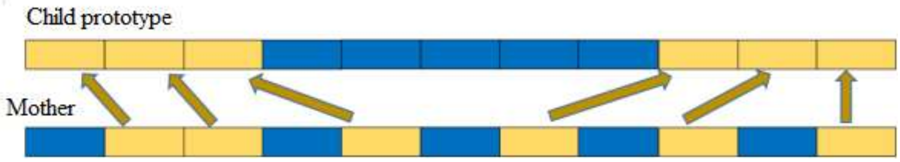

Figure 3. Parent chromosomes generated offspring.

Step 1: During crossover, two chromosomes P1 and P2 are randomly selected each time according to the selection operation as the male and female parent.

Step 2: One chromosome is selected as the paternal parent, and then gene fragment is intercepted in the paternal parent by randomly determining two positions in the chromosome, and the intercepted gene fragment is used as a progeny chromosome prototype. The operation process is shown in Figure 3.

Step 3: The remaining chromosome are taken as the parent, and then completing the missing codes of the progeny prototype from the mother, as shown in Figure 4.

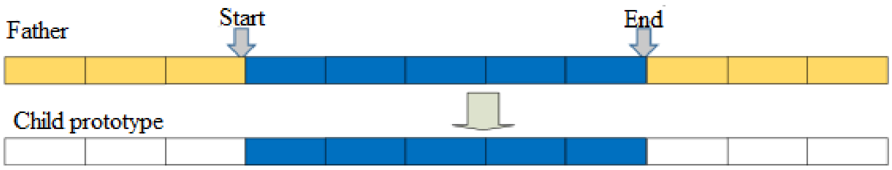

Figure 4. Mother chromosome completes the progeny chromosome prototype.

Step 4: The first child is generated through Steps 2 and 3 above, and then the second child is generated by repeating Steps 2 and 3 above.

The crossover probability is Pc, and the probability can be adjusted adaptively by introducing the number of iterations. The crossover probability is closely related to individual fitness and the number of iterations. The calculation formula of the probability is as follows:

$$P_c = \begin{cases} k_1 \times \dfrac{1}{\sqrt[3]{\text{gen}}} \times \dfrac{f_{\max} - f'}{f_{\max} - f_{avg}} \ldots\ldots\ldots\ldots\left(f' \geq f_{avg}\right) \\ k_1 \times \dfrac{1}{\sqrt[3]{gen}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\left(f' < f_{avg}\right) \end{cases} \quad (5)$$

where $f_{\max}$ is the maximum fitness value in the population; f' is the larger fitness value of the two chromosomes that need to be crossed; f_{avg} is the average value of fitness of all individuals in the population; *gen* is the number of iterations of the population so far; k_1 is a fixed size adjustment parameter ranging from 0 to 1.

3.5. Mutation Operation

Mutation operation has a relatively small probability in the genetic algorithm, but it also has a great impact on the more diverse population. Therefore, the mutation probability should be adjusted slightly in the same way as crossover operation, the mutation probability has a great relationship with the fitness value of chromosomes and the number of iterations of the population. The mutation operation is mainly based on the location mutation method, which randomly selects chromosomes in the population according to the probability, randomly determines the two locations of chromosomes and carries out gene exchange, so as to generate new chromosomes in this way. In this algorithm, two pairs of genes of chromosomes are exchanged, as shown in Figure 5.

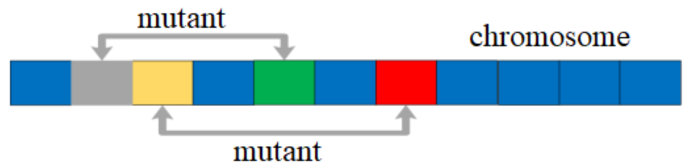

Figure 5. Chromosomal mutations.

The probability of mutation operation is also adjusted adaptively as the number of iterations increases, and the adjustment formula is shown as follows:

$$P_m = \begin{cases} k_2 \times \dfrac{1}{\sqrt{gen}} \times \dfrac{f_{max} - f}{f_{max} - f_{avg}} \text{............} \left(f \geq f_{avg} \right) \\ k_2 \times \dfrac{1}{\sqrt{gen}} \text{..............................} \left(f < f_{avg} \right) \end{cases} \tag{6}$$

where, f is the fitness value of individuals who may be mutated at present; k_2 is the adjustment parameter of individual mutation, generally between (0, 1). Other variable names in the formula have the same meanings as described in (4).

3.6. Simulated Annealing Operator

In order to improve the local searching ability of genetic algorithm, a new hybrid genetic algorithm is proposed. A simulated annealing algorithm is created, which is different from the previous genetic simulated annealing algorithm in that the simulated annealing operator is adjusted. In the past hybrid genetic algorithms, the simulated annealing operator lacks the memory function, But, in this work, using memory function saves the optimal solution generated in the annealing process. The optimal value is determined by a comparison in simulated annealing operator, which is not applicable to the algorithm because the optimal value is highly likely to be ignored. For this IHGA, the optimal value is searched from the population for many times at each temperature state, and the optimal value in this round is determined with a certain probability by reaching the specified optimization times. At temperature T_k (k is the number of cooling times), the optimal value in the population is compared with the randomly selected value, T_1 is the initial temperature, others have similar meanings in turn, and T_{min} is the end temperature, which is set to 0.001 in this work. The cooling formula added in this work as follows:

$$T_{k+1} = \alpha T_k \tag{7}$$

where α is the cooling coefficient.

In the simulated annealing process, in order to prevent the temperature from falling too fast at the initial stage of cooling, appropriate heating strategy is adopted in the annealing process, which is beneficial to increase the acceptance probability of various chromosomes, ensure more diverse selectivity of chromosomes, and avoid the occurrence of local optimization. The search steps are as follows:

Step 1: The current state bit is set as S, the initial value of the cycle counter is $d = 1$, the initial value of the counter for the new individual of the generation population is $O = 1$, and the length of the Markov chain is L.

Step 2: Select v individuals with the lowest fitness value from the population generated after mutation operation as the initial solution in annealing operation, and make the current state bit $S = v$.

Step 3: Select an individual randomly from the population, determine this state bit as $S' = v'$, and calculate the increment of fitness value as $dE = f(v') - f(v)$.

Step 4: If $dE < 0$, the current status bit is determined as S'; Otherwise, the $\exp(-dE/T_K)$ is taken as the probability of S', and $S = S'$.

Step 5: Select the current chromosome for mutation, determine the new state of the mutant chromosome $S'' = v''$, and calculate the fitness value of the new chromosome. If $f(v'') \leq f(v')$, the state bit of the mutant chromosome is S'', that is, $S = S''$; Otherwise, $S = S'$,

and go to Step 3; If $d > L$ is true, the current internal temperature cycle is terminated and Step 6 is performed instead.

Step 6: Calculate and compare the fitness values of all new individuals after the completion of the internal cycle. Then, select the individuals with the lowest fitness values to the new population, and take them as the initial solution of the temperature of the next iteration and make $O = O + 1$.

Step 7: The temperature value of the next iteration is obtained through cooling operation. The calculation formula of temperature difference is $\Delta T = T_K - T_{K+1}$. If $\Delta T > 0.5(T_1 - T_2)$, the heating operation is carried out and $T_{K+1} = T_{K+1} + 0.5\Delta T$.

Step 8: Repeat Step 3 ~ Step 7 until the number of generated individuals reaches the population value or the minimum temperature, and the operation is terminated.

3.7. Operation Process

By analyzing the operation process of the improved hybrid genetic algorithm (IHGA), the operation process is simply described as shown in Figure 6.

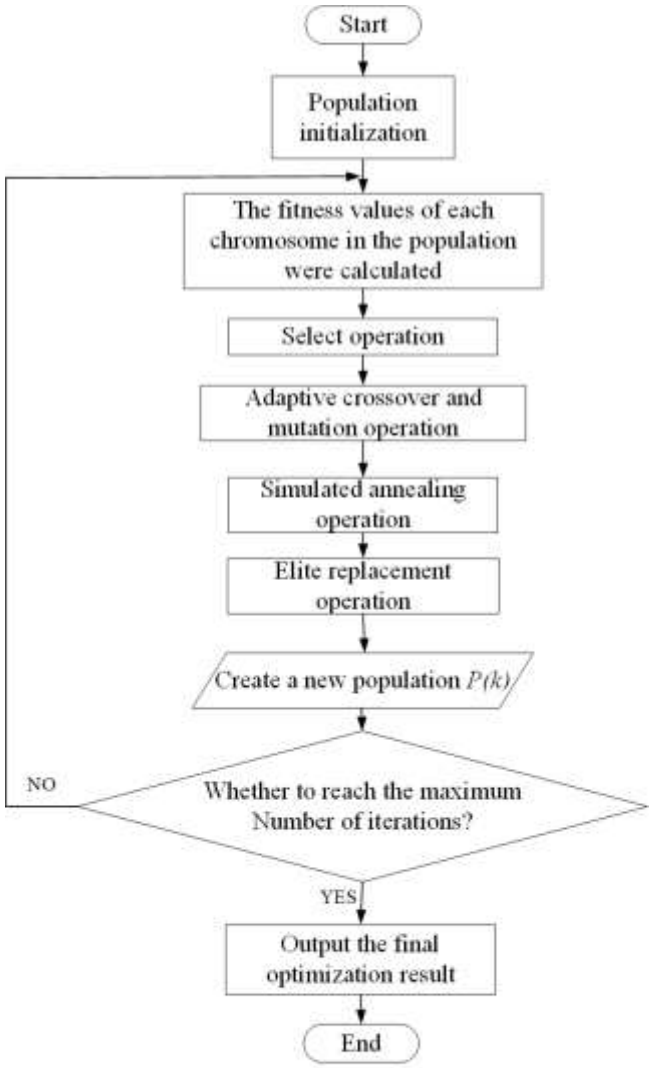

Figure 6. Flowchart of improved hybrid genetic algorithm operation.

4. Experimental analysis

To verify the effectiveness of IHGA in solving production scheduling problems in discrete industries, the method is compared with the improved particle swarm optimization algorithm proposed by Liu Hongming et al. [32], the Quantum Whale optimization algorithm proposed by Yan Xu et al. [33], and traditional genetic algorithm. The FT series and LA series related test case sets are used for comparison. IHGA is written

in C++ language and runs in Visual Studio 2019 software. The operating environment is Windows10 system, the main frequency is 2.60 GHz, and the memory is 8 G personal laptop.

The parameters of the IHGA are set as follows: population size N = 200, evolution algebra G = 100, initial temperature T_1=1000, ending temperature T_{min} = 0.001, cooling coefficient α = 0.98, and Markov chain length L = 200 in simulated annealing.

4.1. Comparison of IHGA and Existing Popular Algorithms

For the discrete shop scheduling problem, the improved particle swarm optimization (IPSO) proposed by Liu et al. [32] and the Quantum Whale optimization algorithm (QWOA) proposed by Yan et al. [33] can deal with some production scheduling problems. But, their optimization performance obviously has some shortcomings compared with the hybrid algorithm proposed in this work. Based on two kinds of algorithm about case set scheduling, comparing with IHGA obtained case set scheduling results, comparison of the two aspects mainly is the average of the scheduling results of optimal solution and the solution of data comparison, as shown in Table 2. The n*m represents the total number of parts and machine; the product of C^* for the optimal solution has been obtained, the Avg. runs the algorithm ten times to get the average of the solutions. According to the comparison results in Table 2, for scheduling case sets FT06, LA01 and LA06, the IHGA algorithm proposed in this work can obtain the known optimal solution due to the combination of the excellent local search ability of simulated annealing operator.

As shown in Table 2, although IHGA did not obtain the known optimal solution for algorithm case set FT10 and FT20, the relevant solutions obtained by IHGA are superior to the other two algorithms, with significantly stronger optimization capability.

Table 2. Comparison of IHGA, QWOA and IPSO

Numerical example	The size is $n*m$	C^*	QWOA		IPSO		IHGA	
			The optimal solution	Avg.	The optimal solution	Avg.	The optimal solution	Avg.
FT06	6*6	55	55	55	55	55	55	55
FT10	10*10	930	983	1045	976	1027	956	982
FT20	20*5	1165	1223	1313	1206	1222	1188	1209
LA01	10*5	666	666	674	666	666	666	666
LA06	15*5	926	926	927	926	926	926	926
LA16	10*10	945	958	1012	973	1011	945	954

4.2. Comparison of IHGA and traditional GA

To further confirm that the improved algorithm is superior to the standard genetic algorithm, the algorithm proposed in this work is compared with the genetic algorithm in searching results, as shown in Table 3.

Table 3. Comparison of IHGA and GA optimization results.

Numerical example	Size (n * m)	C^*	GA		IHGA		Improved effect /%	
			Optimal solution	Avg.	Optimal solution	Avg.	Optimal solution	Avg.
FT06	6*6	55	55	55.4	55	55	0	0.7
FT10	10*10	930	1020	1050	951	990	6.8	5.7
FT20	20*5	1165	1269	1326	1182	1215	6.9	8.4
LA01	10*5	666	666	668	666	666	0	0.2
LA03	10*5	597	597	658	597	609	0	7.5
LA06	15*5	926	926	927	926	926	0	0.1
LA08	15*5	863	863	928	863	870	0	6.3
LA13	20*5	1150	1150	1210	1150	1161	0	4.1
LA18	10*10	848	885	946	848	868	4.2	8.2

Table 4. Details of LA03 datasets.

Process / Artifacts	Each process processing machine and processing time				
	Process 0	Process 1	Process 2	Process 3	Process 4
Artifacts 1	1/23	2/45	0/82	4/84	3/38
Artifacts 2	2/21	1/29	0/18	4/41	3/50
Artifacts 3	2/38	3/54	4/16	0/52	1/51
Artifacts 4	4/37	0/54	2/74	1/62	3/57
Artifacts 5	4/37	0/81	1/61	3/68	2/30
Artifacts 6	4/81	0/79	1/89	2/89	3/11
Artifacts 7	3/33	2/20	0/91	4/20	1/66
Artifacts 8	4/24	1/84	0/32	2/55	3/8
Artifacts 9	4/56	0/7	3/54	2/64	1/39
Artifacts 10	4/40	1/83	0/19	2/8	3/7

It can be seen from Table 3 that in FT06, LA01, LA03, LA06, LA08 and LA13 cases, both algorithms can finally obtain the optimal solution of the scheduling problem, but the average value obtained by IHGA is superior to GA. The IHGA did not find the optimal

solution of scheduling cases in FT10, FT20 and LA18. But compared with GA, the optimization effect was greatly improved, with the improvement effect ranging from 4.2% to 6.9%, and the average improvement effect ranging from 5.7% to 8.4%. In the comparison of the two algorithms, the search result is greatly improved mainly because the simulated annealing operator improves the local search ability of genetic algorithm, which greatly improves the search ability of the algorithm proposed in this work.

Taking LA03 in the test datasets as an example, the feasibility and superiority of the algorithm proposed in this work in solving discrete scheduling problems are explained in detail. The specific datasets of LA03 are shown in Table 4. The data in Table 4 represents that processing parts by a process uses serial number of the machine and the processing time, for example the corresponding parts in the Table 2, the 1/29 means in the processing of selecting the Numbers for 1 second process machinery for processing, it takes time for 29, and other data has the same meaning.

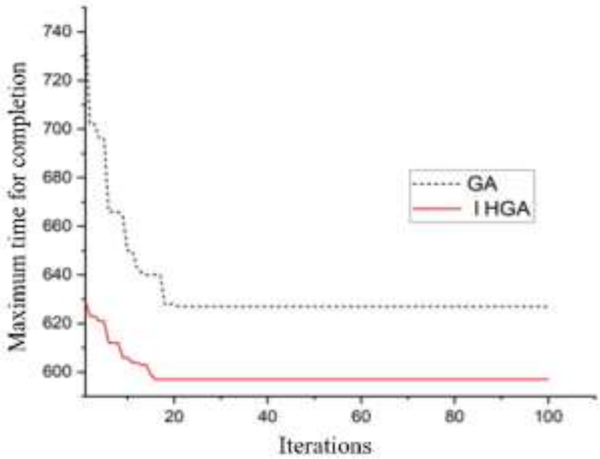

Figure 7. LA03 convergence curve.

For example, it can be seen from the comparison of results in Table 3 that both IHGA and GA algorithms can find the optimal solution, but the average value obtained by IHGA is better than that obtained by GA, which means that IHGA has good robustness. After a period of comparative testing, the convergence of the iterative curves of the two algorithms in dealing with the LA03 scheduling problem is shown in Figure 7. As can be seen from Figure 7, GA not only has a slow convergence speed when searching for the optimal value, but also cannot find the optimal value, and converging at about 630. The IHGA has a relatively fast convergence speed, and can jump out of the local optimum in time and find the global optimum value of 597. This indicates that the IHGA proposed in this work is obviously superior to the genetic algorithm, and can solve the production scheduling problem of the workshop more effectively.

Taking LA03 for example, the IHGA can work out the currently known optimal fitness solution 597, select the chromosome with the best fitness, and obtain the corresponding scheduling Gantt chart according to the process arrangement results of each part in the chromosome, as shown in Figure 8. The Gantt chart can prove that the IHGA proposed in this work can provide an excellent and effective solution for production scheduling in discrete industries.

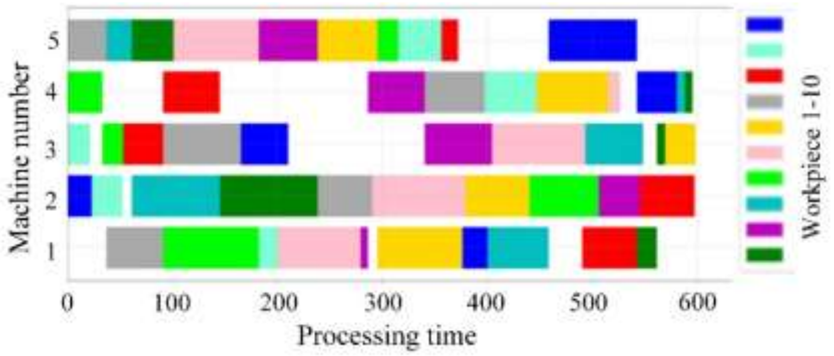

Figure 8. Scheduling Gantt diagram of LA03 obtained by the IHGA,

5. Industrial Field Test

A MES system is developed by C# object-oriented development language, and the corresponding database is established by SQL server, where production scheduling function adopts the proposed IHGA. To verify the feasibility of the MES system, industrial application tests were carried out with the help of a network environment and hardware platform of discrete manufacturing industry. Reducer production project of the industry was selected as the basis, and the business process of the designed MES system was verified in workshop. The specific steps are as follows:

Step 1: Production plan. According to order demands, production planner make specific reducer production project, where the 12 kinds of parts are set in production process, and the project is distributed after the process and working hours of each part are modified. The system records the names of the employees who distribute and receive the production plan. The distributed production plan contains basic information.

Step 2: Production scheduling. The team leader of the production unit checks the distributed production plan to conduct scheduling optimization according to the IHGA, and the optimized results through Gantt chart is displayed as shown in Figure 10. The processing sequence and corresponding processing time of different parts on different equipment are showed, and further optimization of the production plan has been realized, including specific production time and operating staff.

Step 3: Production and processing. Workshop operators receive the production plan distributed by the team leader, logging in the MES system according to their own authority and obtaining the parts to be produced. Then, the parts are produced according to the time of production process and process drawings. The system records the name and production time of employees, and the production team leader using the system can see the production status of the process at any time, as shown in Figure 9.

Figure 9. Gear shop operation diagram.

The industrial experiment proves that the IHGA can realize the production scheduling of the industry, so that the production of a variety of products can be completed in a relatively short time, and greatly improve the production efficiency of the workshop. According to the detailed production plan generated after scheduling, the on-site production situation of the workshop is shown in Figure 10. It can be further seen that when the operator uses drilling machine and lathe to process the reducer workpiece, the optimized software can run normally and record the production time, equipment name and employee name to reflect the processing status of the parts and facilitate the traceability of defective products. Real-time display is carried out on the large screen of the workshop to reflect the working efficiency of the staff, the running status of the equipment and the production situation, etc., and the transparent production of the workshop is realized.

Figure 10. Production scene diagram of discrete manufacturing industry.

5. Conclusions

Firstly, the basic situation of production scheduling in discrete industries is introduced in this paper. Considering that simulated annealing algorithm can improve the shortcomings of genetic algorithm optimization, an improved hybrid genetic algorithm is proposed to solve the problem of production scheduling. In this work, the shortcomings of genetic algorithm are analyzed and studied, and the adaptive strategy is adopted to adjust the probability of crossover and variation of genetic operators. The adjustment mainly depends on the number of iterations and fitness, so that the population diversity is high in the early stage, and convergence is possible in the later stage. For the added simulated annealing operator, the elitist substitution strategy is adopted to save the optimal solution and avoid the occurrence of local optimal of genetic algorithm. The results show that the improved hybrid genetic algorithm has better optimization ability compared with other scheduling algorithms. Compared with the genetic algorithm, although the optimal solution was not obtained in the three cases, the improvement effect of the optimal solution was 4.2% ~ 6.9%, and the average improvement range was 5.7% ~ 8.4%. The simulated annealing operator played a good local optimization effect in IHGA, which greatly improved the optimization ability of the overall algorithm.

In the actual production process of discrete industries, production scheduling considers more complex indexes, so the next main task is to study the flexible production scheduling with multiple indexes, improve the production efficiency of workshop, reduce resource waste.

Acknowledgements

The research leading to these results has received funding from the Norwegian Financial Mechanism 2014-2021 under Project Contract No 2020/37/K/ST8/02748.

References

1. Wang P., Xiao X. C., Yuan L. H., Fang J.. Application and Research in Discrete Manufacturing Industry Based on MES, China Science and Technology Information, 2015 (18): 119-120.
2. Yang N.. Implementation and Application of Discrete Manufacturing MES System, Equipment Manufacturing Technology, 2020 (05): 273-275+278.
3. Jin B. F., Li B.. Multi Objective Job Shop Scheduling Based on Genetic Algorithm, Modular Machine Tools and Automatic Processing Technology, 2019 (4): 4.
4. Cao R., Hou X. P., Jin S. T.. Research on Flexible Job Shop Scheduling Problem Based on Improved Genetic Algorithm, Computer and Digital Engineering, 2019, 47 (2): 4.
5. Lu y., Feng Z. J., Hu Y.. Research on Scheduling Algorithm for Multiple Handling Requests of A Single AGV Car, Modular Machine Tools and Automatic Processing Technology, 2019 (2): 4.
6. Kumar N., Mishra A.. Comparative Study of Different Heuristics Algorithms in Solving Classical Job Shop Scheduling Problem, Materials Today: Proceedings, 2020, 22: 1796-1802.
7. Kong L., Wang L.M., Li F. i.. Green Production Scheduling of Hybrid Flow Shop Based on Machining Matching Characteristics of Machine Tool, Computer Integrated Manufacturing System, 2019, 25 (5): 11.
8. Xing G., Nyberg T. R.. Comparison Between CIMS and CIPS, IEEE International Conference on Emerging Technologies & Factory Automation. IEEE, 1999: 409-416.
9. Chen X. H.. Research on Reconfigurable Architecture and Key Technologies of Discrete Manufacturing Execution System Based on Semantic Gateway, Chongqing: Chongqing University, 2014.
10. Chang J. T.. Research on Some Key Technologies in Production Management of Discrete Manufacturing Enterprises, Xi'an: Xi'an University of Electronic Science and technology, 2014.
11. Zhong R. Y., Huang G. Q., Dai Q. Y., et al. RFID-enabled Real-time Manufacturing Execution System for Discrete Manufacturing: Software Design and Implementation, IEEE International Conference on Networking. IEEE Xplore, 2011: 311-316.
12. Li F., Li X. H.. Application Practice of Manufacturing Execution System in Discrete Industries, Intelligent Manufacturing, 2008 (1): 24-27.
13. Bian D. Z., Guo J. M., Wang M., Song F. F.. Research on MES System Based on Discrete Shipbuilding, Manufacturing Technology and Machine Tools, 2020 (01): 149-152.
14. Mao K., Pan Q. K., Pang X. F.. Lagrangian Algorithm for Steelmaking Continuous Casting Production Scheduling Problem, Journal of Systems Engineering, 2014, 29 (2): 13.
15. Bellabdaoui A, Teghem J. A Mixed-integer Linear Programming Model for The Continuous Casting Planning, International Journal of Production Economics, 2006, 104(2): 260-270.
16. BELLMAN R. E., DREYFUS S. E.. Applied Dynamic Programming [M]. [S.1]: Princeton University Press, 2015.
17. Wu D. W., Lu T. T., Liu X. B., et al. Parallel Simulated Annealing Algorithm for Job Shop Scheduling Problem, Computer Integrated Manufacturing System, 2005, 11 (6): 847-850.
18. Zhang J., Wang W. L., Xu X. L.. et al Hybrid Particle Swarm Optimization Algorithm for Multi-objective Flexible Job Shop Scheduling Problem, Control Theory and Application, 2012, 29 (6): 8.
19. Wang W. L., Zhao C., Xing J.. A Method for Solving Flexible Job Shop Scheduling Problem Based on Improved ant Colony Algorithm, Journal of System Simulation, 2008, 20 (16): 4.
20. Choi I. C., Choi D. S.. A Local Search Algorithm for Job Shop Scheduling Problems with Alternative Operations and Sequence-dependent Setups, Computers & Industrial Engineering, 2002, 42(1): 43-58.
21. Nie L., Gao L., Li P., et al. A GEP-Based Reactive Scheduling Policies Constructing Approach for Dynamic Flexible Job Shop

Scheduling Problem with Job Release Dates, Journal of Intelligent Manufacturing, 2013, 24(4): 763-774.

22. Chen J. C., Chen K. H., Wu J. J., et al. A Study of The Flexible Job Shop Scheduling Problem with Parallel Machines and Reentrant Process, The International Journal of Advanced Manufacturing Technology, 2008, 39(3-4): 344-354.
23. Lu T. T., Chen P., Wan X. Y.. An Improved Cellular Genetic Algorithm for Solving Multi-objective Flexible Job Shop Scheduling Problem, Modern Manufacturing Engineering, 2016, 000 (011): 41-49.
24. Gao J., Gen M., Sun L., et al. A Hybrid of Genetic Algorithm and Bottleneck Shifting for Multi-objective Flexible Job Shop Scheduling Problems, Computers & Industrial Engineering, 2007, 53(1): 149-162.
25. Pezzella F., Morganti G., Ciaschetti G.. A Genetic Algorithm for The Flexible Job-shop Scheduling Problem, Computers & Operations Research, 2008, 35(10): 3202-3212.
26. De Giovanni L., Pezzella F.. An Improved Genetic Algorithm for The Distributed and Flexible Job-shop Scheduling Problem, European Journal of Operational Research, 2010, 200(2): 395-408.
27. Zhang R., Chang P. C., Wu C.. A Hybrid Genetic Algorithm for The Job Shop Scheduling Problem with Practical Considerations for Manufacturing Costs: Investigations Motivated by Vehicle Production, International Journal of Production Economics, 2013, 145(1): 38-52.
28. Chamnanlor. Embedding Ant System in Genetic Algorithm for Re-entrant Hybrid Flow Shop Scheduling Problems with Time Window Constraints, Journal of Intelligent Manufacturing, 2015, 40(10): 1-17.
29. Meng Y., Zhao S. H.. Solving Job Shop Scheduling Problem with Hybrid Algorithm Integrating Path Reconnection, Mechanical Engineer, 2019 (12): 32-36+39.
30. Zhao W.. Application of Simulated Annealing Genetic Glgorithm in Job Shop Scheduling, Computer Simulation, 2011, 28 (07): 361-364.
31. Wang J. H., Lu C.. Application of A Genetic Algorithm Combined with Simulated Annealing in Flexible Job Shop Scheduling, Digital Technology and Application, 2019, 37 (01): 133-136.
32. Liu H. M., Zeng H. Y., Zhou W., Wang T.. Optimization of Job Shop Scheduling Problem Based on Improved Particle Swarm Optimization, Journal of Shandong University (Engineering Edition), 2019, 49 (01): 75-82.
33. Yan X., Ye C. M., Yao Y.. Quantum Whale Optimization Algorithm for Job Shop Scheduling, Computer Application Research, 2019, 36 (04): 975-979.